Set It and Forget It: Approximating the Set Once Strip Cover Problem

Amotz Bar-Noy[1], Ben Baumer[1], and Dror Rawitz[1,2]

[1] The Graduate Center of the City University of New York
amotz@sci.brooklyn.cuny.edu, bbaumer@gc.cuny.edu
[2] School of Electrical Engineering, Tel Aviv University
rawitz@eng.tau.ac.il

Abstract. We consider the Set Once Strip Cover problem, in which n wireless sensors are deployed over a one-dimensional region. Each sensor has a fixed battery that drains in inverse proportion to a radius that can be set just once, but activated at any time. The problem is to find an assignment of radii and activation times that maximizes the length of time during which the entire region is covered. We show that this problem is NP-hard. Second, we show that RoundRobin, the algorithm in which the sensors simply take turns covering the entire region, has a tight approximation guarantee of $\frac{3}{2}$ in both Set Once Strip Cover and the more general Strip Cover problem, in which each radius may be set finitely-many times. Moreover, we show that the more general class of *duty cycle* algorithms, in which groups of sensors take turns covering the entire region, can do no better. Finally, we give an optimal $O(n^2 \log n)$-time algorithm for the related Set Radius Strip Cover problem, in which all sensors must be activated immediately.

1 Introduction

Suppose that n sensors are deployed over a one-dimensional region that they are to cover with a wireless network. Each sensor is equipped with a finite battery charge that drains in inverse proportion to the sensing radius that is assigned to it, and each sensor can be activated only once. In the Set Once Strip Cover (OnceSC) problem, the goal is to find an assignment of radii and activation times that maximizes the *lifetime* of the network, namely the length of time during which the entire region is covered.

Formally, we are given as input the locations $x \in [0,1]^n$ and battery charges $b \in \mathbb{Q}^n$ for each of n sensors. While we cannot move the sensors, we do have the ability to set the sensing radius ρ_i of each sensor and the time τ_i when it should become active. Since each sensor's battery drains in inverse proportion the radius we set (but cannot subsequently change), each sensor covers the region $[x_i - \rho_i, x_i + \rho_i]$ for b_i/ρ_i time units. Our task is to devise an algorithm that finds a schedule $S = (\rho, \tau) \in [0,1]^n \times [0,\infty)^n$ for any input (x, b), such that $[0,1]$ is completely covered for as long as possible.

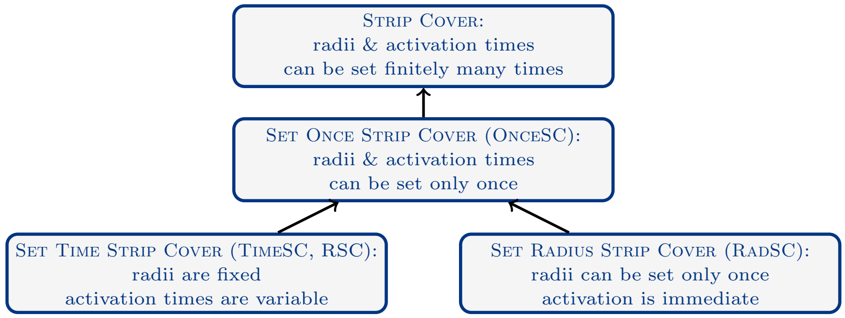


Fig. 1. Relationship of Problem Variants.

Motivation. This type of scheduling problem arises in many applications. Suppose that we have a highway, supply line, or fence in territory that is either hostile or difficult to navigate. While we want to monitor activity along this line, conditions on the ground make it impossible to systematically place wireless sensors at specific locations. However, it is feasible and inexpensive to deploy adjustable range sensors along the line by, say, dropping them from an airplane flying overhead. Once deployed, the sensors send us their location via GPS, and we wish to send a single radius-time pair to each sensor as an assignment. Replacing the battery in any sensor is infeasible. How do we construct an assignment that will keep this vital supply line completely monitored for as long as possible?

Models. While the focus of this paper is the OnceSC problem, we touch upon three closely related problems. In each problem the location and battery of each sensor are fixed, and a solution can be viewed as a finite set of radius-time pairs. In OnceSC, both the radii and the activation times are variable, but can be set only once. In the more general Strip Cover problem, the radius and activation time of each sensor can be set finitely many times. On the other hand, if the radius of each sensor is fixed and given as part of the input, then we call the problem of assigning activation time to each sensor so as to maximizes network lifetime Set Time Strip Cover (TimeSC). Set Radius Strip Cover (RadSC) is another variant of OnceSC in which all of the sensors are scheduled to activate immediately, and the problem is to find the optimal radial assignment. Figure 1 summarizes the important differences between related problems and illustrates their relationship to one another.

Related work. The Strip Cover problem was first considered by Bar-Noy and Baumer [2], who gave a $\frac{3}{2}$ lower bound on the performance of RoundRobin (see Obs. 2), but were only able to show a corresponding upper bound of 1.82. The similar Connected Range Assignment (CRA) problem, in which radii are assigned to points in the plane in order to obtain a connected disk graph, was studied by Chambers, et al. [5]. They showed that the best one circle solution to CRA also yields a $\frac{3}{2}$-approximation guarantee, and in fact, the instance that produces their lower bound is simply a translation of the instance used in Obs. 2.

The TimeSC problem, which is known as Restricted Strip Covering (RSC), was shown to be NP-hard by Buchsbaum, et al., who also gave an $O(\log \log n)$-approximation algorithm [4]. Later, a constant factor approximation algorithm was discovered by Gibson and Varadarajan [6].

Close variants of RadSC have been the subject of previous work. Whereas RadSC requires *area* coverage, Peleg and Lev-Tov [7] studied *target* coverage. In this problem the input is a set of n sensors and a finite set of m points on the line that are to be covered, and the goal is to find the radial assignments with the minimum sum of radii. They used dynamic programming to devise a polynomial time alorithm. Bar-Noy, et al. [3] improved the running time to $O(n + m)$.

The notion of *duty cycling* as a means to maximize network lifetime first appeared in the literature of discrete geometry. In this context, maximizing the number of covers k serves as a proxy for maximizing the actual network lifetime. Pach [8] began the study of decomposability of multiple coverings. Pach and Tóth [9] showed that a k-fold cover of translates of a centrally-symmetric open convex polygon can be decomposed into $\Omega(\sqrt{k})$ covers. This was later improved to the optimal $\Omega(k)$ covers by Aloupis et al. [1], while Gibson and Varadarajan [6] showed the same result without the centrally-symmetric restriction.

Our contributions. We introduce the Set Once model that corresponds to the case where the scheduler does not have the ability to vary the sensor's radius once it has been activated. We show that OnceSC is NP-hard (Appendix A), and that RoundRobin is a $\frac{3}{2}$-approximation algorithm for both OnceSC (Section 3) and Strip Cover (Section 5). This closes a gap between the best previously known lower ($\frac{3}{2}$) and upper (1.82) bounds for the performance of this algorithm. Our analysis of RoundRobin is based on comparing its performance to the RadSC optimum of certain instances with unit batteries. Moreover, in Appendix B we show that the larger class of duty cycle algorithms cannot improve on this $\frac{3}{2}$ guarantee. In Section 4, we provide an $O(n^2 \log n)$-time algorithm for RadSC. Finally, we note that several proofs and figures were relegated to the appendix due to space considerations.

2 Preliminaries

Problems. The Set Once Strip Cover (abbreviated OnceSC) is defined as follows. Let $U = [0, 1]$ be the interval that we wish to cover. Given is a vector $x = (x_1, \ldots, x_n) \in U^n$ of n sensor locations, and a corresponding vector $b = (b_1, \ldots, b_n) \in \mathbb{Q}_+^n$ of battery charges, with $b_i \geq 0$ for all i. We assume that $x_i \leq x_{i+1}$ for every $i \in \{1, \ldots, n-1\}$. We sometimes abuse notation by treating x as a set. An instance of the problem thus consists of a pair $I = (x, b)$, and a solution is an assignment of radii and activation times to sensors. More specifically a solution (or *schedule*) is a pair $S = (\rho, \tau)$ where ρ_i is the *radius* of sensor i and τ_i is the *activation time* of i. Since the radius of each sensor cannot be reset, this means that sensor i becomes active at time τ_i, covers the *range* $[x_i - \rho_i, x_i + \rho_i]$ for b_i/ρ_i time units, and then becomes inactive since it has exhausted its entire battery.

Any schedule can be visualized by a space-time diagram in which each coverage assignment can be represented by a rectangle. It is customary in such diagrams to view the sensor locations as forming the horizontal axis, with time extending upwards vertically. In this case, the coverage of a sensor located at x_i and assigned the radius ρ_i beginning at time τ_i is depicted by a rectangle with lower-left corner $(x_i - \rho_i, \tau_i)$ and upper-right corner $(x_i + \rho_i, \tau_i + b_i/\rho_i)$. Let the set of all points contained in this rectangle be denoted as $Rect(\rho_i, \tau_i)$. A point (u, t) in space-time is *covered* by a schedule (ρ, τ) if $(u, t) \in \bigcup_i Rect(\rho_i, \tau_i)$. The *lifetime* of the network in a solution $S = (\rho, \tau)$ is the maximum value T such that every point $(u, t) \in U \times [0, T]$ is covered. Graphical depications of two schedules are shown below in Figure 2.

In OnceSC our goal is to find a schedule $S = (\rho, \tau)$ that maximizes the lifetime T. Given an instance $I = (x, b)$, the optimal lifetime is denoted by $\text{Opt}(x, b)$. (We sometimes use Opt, when the instance is clear from the context.)

The Set Radius Strip Cover (RadSC) problem is a variant of OnceSC in which $\tau = 0^n$. Hence, a solution is simply a radial assignment ρ. Set Time Strip Cover (TimeSC) is another variant in which the radii are given in the input, and a solution is an assignment of activation times to sensors.

Strip Cover is a generalization of OnceSC in which a sensor's radius may be changed finitely many times. In this case a solution is a vector of piece-wise constant functions $\rho(t)$, where $\rho_i(t)$ is the sensing radius of sensor i at time t. The solution is feasible if U is covered for all $t \in [0, T]$, and if $\int_0^\infty \rho_i(t)\, dt \le b_i$, for every i. The segment $[0, 1]$ is covered at time t, if $[0, 1] \subseteq \bigcup_i [x_i - \rho_i(t), x_i + \rho_i(t)]$.

Maximum lifetime. The best possible lifetime of an instance (x, b) is $2 \sum_i b_i$. We state this formally for OnceSC, but the same holds for the other variants.

Observation 1. *The lifetime of a* OnceSC *instance* (x, b) *is at most* $2 \sum_i b_i$.

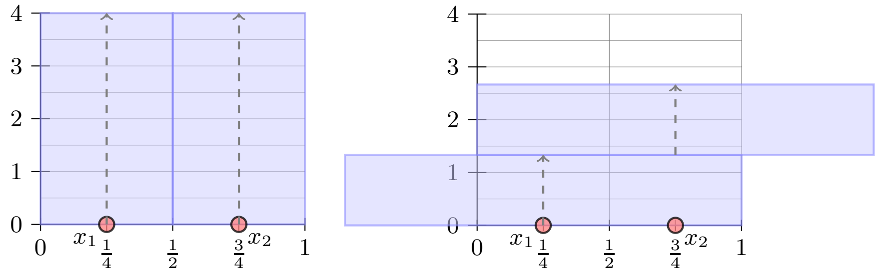
Proof. Consider an optimal solution (ρ, τ) for (x, b) with lifetime T. A sensor i covers an interval of length $2\rho_i$ for b_i/ρ_i time. The lifetime T is at most the total area of space-time covered by the sensors, which is at most $\sum_i 2\rho_i \cdot b_i/\rho_i$. □

Round Robin. We focus on a simple algorithm we call RoundRobin. The RoundRobin algorithm forces the sensors to take turns covering U, namely it assigns, for every i, $\rho_i = r_i \triangleq \max\{x_i, 1 - x_i\}$ and $\tau_i = \sum_{j=1}^{i-1} b_j/\rho_j$. The lifetime of RoundRobin is thus

$$\text{RR}(x, b) \triangleq \sum_{i=i}^{n} \frac{b_i}{r_i} \ .$$

A lower bound of $\frac{3}{2}$ on the approximation guarantee of RoundRobin was given in [2] using the two sensor instance $x = (\frac{1}{4}, \frac{3}{4})$, $b = (1, 1)$. The relevant schedules are depicted graphically in Figure 2.

Observation 2 ([2]). *The approximation ratio of* RoundRobin *is at least* $\frac{3}{2}$.



(a) $\text{OPT}(x,b) = 4$ (b) $\text{ROUNDROBIN}(x,b) = 2\frac{2}{3}$

Fig. 2. ROUNDROBIN vs. OPT with $x = (\frac{1}{4}, \frac{3}{4})$, and $b = (1,1)$.

Given an instance (x,b) of ONCESC, let $B \triangleq \sum_i b_i$ be the total battery charge of the system and $\overline{r} = \sum_i \frac{b_i}{B} \cdot r_i$ be the average of the r_i's, weighted by their respective battery charge. We define the following lower bound on $\text{RR}(x,b)$:

$$\text{RR}'(x,b) \triangleq B/\overline{r} \ .$$

Lemma 1. $\text{RR}'(x,b) \le \text{RR}(x,b)$*, for every* ONCESC *instance* (x,b)*.*

Proof. $\text{RR}(x,b) = \sum_{i=1}^{n} \frac{b_i}{r_i} = \sum_{i=1}^{n} \frac{b_i^2}{b_i r_i} \ge \frac{(\sum_{i=1}^{n} b_i)^2}{\sum_{i=1}^{n} b_i r_i} = \frac{\sum_{i=1}^{n} b_i}{\overline{r}} = \text{RR}'(x,b)$, where the inequality is due to the following implication of the Cauchy-Schwarz Inequality: for any positive $c, d \in \mathbb{R}^n$, it holds that $\sum_j \frac{c_j^2}{d_j} \ge \frac{(\sum_j c_j)^2}{\sum_j d_j}$. $\square$

3 Round Robin

We show in Appendix A that ONCESC is NP-hard, so we turn our attention to approximation algorithms. While ROUNDROBIN is among the simplest possible algorithms (note that its running time is exactly n), the precise value of its approximation ratio is not obvious, although it is not hard to see that 2 is an upper bound. In [2] an upper bound of 1.82 and a lower bound of $\frac{3}{2}$ were shown. In this section, we show that the approximation ratio of ROUNDROBIN in ONCESC is exactly $\frac{3}{2}$.

The structure of the proof is as follows. We start with an optimal schedule S, and cut it into disjoint time intervals, or strips, such that the same set of sensors is active within each time interval. Each strip induces a RADSC instance I_j and solution S_j. For each such strip $I_j = (x^j, b^j)$ we compare the performance of ROUNDROBIN to the lifetime T_j of the strip solution S_j. More specifically, we prove that $\text{RR}(x^j, b^j) \ge \frac{3}{2} T_j$. We do this using a reduction to a special case of RADSC with unit batteries.

3.1 Cutting the Schedule into Strips

Given an instance $I = (x,b)$, and a solution $S = (\rho, \tau)$ with lifetime T, let Ω be the set of times until T in which a sensor was turned on or off, namely

$\Omega = \bigcup_i \{\tau_i, \tau_i + b_i/\rho_i\} \cap [0,T]$. Let $\Omega = \{\omega_1, \ldots, \omega_\ell\}$, where $\omega_j < \omega_{j+1}$, for every j. Notice that $0, T \in \Omega$. Next, we partition the time interval $[0,T]$ into the sub-intervals $[\omega_j, \omega_{j+1}]$, for every $j \in \{1, \ldots, \ell-1\}$.

We define a new instance for every sub-interval. Let $x^j \subseteq x$, for every $j \in \{1, \ldots, \ell-1\}$ be the set of sensors that participate in covering $[0,1]$ during the jth sub-interval. That is, $x^j = \{x_i : [\omega_j, \omega_{j+1}] \subseteq [\tau_i, \tau_i + b_i/\rho_i]\}$. Also, let $T_j = \omega_{j+1} - \omega_j$, and let b_i^j be the energy that was consumed by sensor i during the jth sub-interval, i.e., $b_i^j = \rho_i \cdot T_j$. Observe that $I_j = (x^j, b^j)$ is a valid instance of RadSC, for which ρ^j, where $\rho_i^j = \rho_i$ for every sensor i such that $x_i \in x^j$, is a solution that achieves a lifetime of exactly T_j.

We further modify the instance $I_j = (x^j, b^j)$ and the solution ρ^j as follows:

- Starting with $i = 1$, we remove sensor i from the instance if the interval $[0,1]$ is covered during $[\omega_j, \omega_{j+1}]$ without i.
- We decrease the battery and the radius of the left-most sensor as much as possible, and we also decrease the battery and the radius of the right-most sensor as much as possible.

Figure 5 in Appendix C provides an illustration of this procedure.

Observation 3. *Let sensors* 1 *and* m *be the left-most and right most sensors in* x^j. *Then, either* $\rho_1^j = x_1$ *or the point* $x_1^j + \rho_1^j$ *is only covered by sensor* 1. *Also, either* $\rho_m^j = 1 - x_m$ *or the point* $x_m^j - \rho_m^j$ *is only covered by sensor* m.

Observe that $\mathrm{RR}(x^j, b^j) = \sum_{x_i \in x^j} \frac{b_i^j}{r_i}$ is the RoundRobin lifetime of the jth strip. In the sequel we show that

Lemma 2. $\mathrm{RR}'(x^j, b^j) \geq \frac{2}{3} T_j$, *for every* j.

It follows that

Theorem 1. RoundRobin *is a* $\frac{3}{2}$*-approximation algorithm for* OnceSC.

Proof. First, observe that

$$\sum_j \mathrm{RR}(x^j, b^j) = \sum_j \sum_{x_i \in x^j} \frac{b_i^j}{r_i} = \sum_i \frac{1}{r_i} \sum_{j : x_i \in x^j} b_i^j \leq \sum_i \frac{1}{r_i} b_i = \mathrm{RR}(x, b) \ .$$

By Lemmas 1 and 2 we have that $\mathrm{RR}(x,b) \geq \sum_j \mathrm{RR}(x^j, b^j) \geq \sum_j \mathrm{RR}'(x^j, b^j) \geq \sum_j \frac{2}{3} T_j = \frac{2}{3} T$. □

3.2 Reduction to Set Radius Strip Cover with Uniform Batteries

Given the RadSC instance $I_j = (x^j, b^j)$ and a solution ρ^j, we construct an instance $I_j' = (y^j, \bar{1})$ with unit size batteries and a RadSC solution σ^j, such that the lifetime of σ_j is T_j.

Let Opt_0 denote the optimal RadSC lifetime. We assume that $b_i^j \in \mathbb{N}$ and $b_i^j \geq 3$ for every i, since (i) $b_i^j \in \mathbb{Q}$ for every i, (ii) $\mathrm{Opt}_0(x, \beta b) = \beta \cdot \mathrm{Opt}_0(x, b)$, and (iii) $\mathrm{RR}(x, \beta b) = \beta \cdot \mathrm{RR}(x, b)$.

The instance I'_j is constructed as follows. We replace each sensor i such that $x_i^j \in x^j$ with b_i^j unit battery sensors whose average location is x_i. These unit battery sensors are called the *children* of i. To do this, we divide the interval $[x_i^j - \rho_i^j, x_i^j + \rho_i^j]$ into b_i^j equal sub-intervals, and place a unit battery sensor in the middle of each sub-interval. Observe that child sensors may be placed outside $[0,1]$, namely to the left of 0 or to the right of 1. The solution σ_j is defined as follows. For any child k of a sensor i in I_j, we set $\sigma_k^j = \rho_i^j / b_i^j$. An example is shown in Figure 6.

We prove that the lifetime of σ^j is T_j.

Lemma 3. *The lifetime of σ_j is T_j.*

Proof. First, the b_i^j children of a sensor i in I_j cover the interval $[x_i^j - \rho_i^j, x_i^j + \rho_i^j]$. Also, a child k of i survives $1/\sigma_k^j = b_i^j / \rho_i^j = T_j$ time units. □

Next, we prove that the lower bound on the performance of RoundRobin did not change.

Lemma 4. $\mathrm{RR}'(y^j, \bar{1}) = \mathrm{RR}'(x^j, b^j)$.

Proof. Let p^j be the RoundRobin radii of y^j. Hence,

$$\mathrm{RR}'(y^j, \bar{1}) = \frac{\sum_i b_i^j}{\overline{p^j}} = \frac{B^j}{\frac{1}{B^j}\sum_k p_k^j} = \frac{B^j}{\frac{1}{B^j}\sum_i b_i^j r_i^j} = \frac{B^j}{\overline{r^j}} = \mathrm{RR}'(x^j, b^j)\,,$$

where we have used the fact that a parent's location is equal to the average location of her children. □

Lemma 2 now follows from Lemmas 3 and 4 and Lemma 8. The latter states that $\mathrm{RR}'(y^j, \bar{1}) \geq \frac{2}{3}\mathrm{OPT}_0(y^j, \bar{1})$, and we devote the next section to proving it.

3.3 Analysis of Round Robin for Unit Batteries

For the remainder of this section, we assume that we are given a unit battery instance x that corresponds to the jth strip. (We drop the subscript j and go back to x for readability.) Recall that $x \cap [0,1]$ is not necessarily equal to x, since some children could have been created outside $[0,1]$. We show that $\mathrm{RR}'(x) \geq \frac{2}{3}\mathrm{OPT}_0(x)$.

Let $i_0 = \min\{i : x_i \geq 0\}$ and let $i_1 = \max\{i : x_i \leq 1\}$ be the indices of the leftmost and rightmost sensors in $[0,1]$, respectively.

Lemma 5. $\max_{i \in \{i_0,\ldots,i_1-1\}}\{x_{i+1} - x_i\} = \max_{i \in \{1,\ldots,n-1\}}\{x_{i+1} - x_i\}$.

Proof. By Observation 3 either $\rho_1^j = x_1$ and hence none of its children are located to the left of 0, or the point $x_1^j + \rho_1^j$ is only covered by sensor 1 which means that the gaps between 1's children to the left of zero also appears between its children within $[0,1]$. (Recall that $b_i^j \geq 3$, for all i.) The same argument can be used for the right-most sensor. □

We define

$$d_0 \triangleq \begin{cases} x_{i_0} - x_{i_0-1} & \text{if } i_0 > 1 \text{ and } -x_{i_0-1} < x_{i_0}\,, \\ 2x_{i_0} & \text{otherwise,} \end{cases}$$

and

$$d_1 \triangleq \begin{cases} x_{i_1+1} - x_{i_1} & \text{if } i_0 < n \text{ and } x_{i_0+1} - 1 < 1 - x_{i_1}\,, \\ 2(1 - x_{i_1}) & \text{otherwise,} \end{cases}$$

and

$$\Delta \triangleq \max\left\{ d_0, d_1, \max_{i \in \{i_0,\ldots,i_1-1\}} \{x_{i+1} - x_i\} \right\} \ .$$

We describe the optimal RadSC lifetime in terms of Δ.

Lemma 6. *The optimum lifetime of x is $\frac{2}{\Delta}$.*

Proof. To verify that $2/\Delta$ can be achieved, consider the solution in which $\rho_i = \Delta/2$ for all i. Clearly, $[0,1]$ is covered, and all sensors die after $2/\Delta$ time units. Now suppose that a solution ρ exists with lifetime strictly greater than $2/\Delta$. Hence $\max_i\{\rho_i\} < \Delta/2$. By definition, Δ must equal d_0, d_n, or the maximum internal gap. If the latter, then there exists a point $u \in [0,1]$ between the two sensors forming the maximum internal gap that is uncovered. On the other hand, if $\Delta = d_0$, then if $d_0 = 2x_{i_0}$, 0 is uncovered, and otherwise, there is a point in $[0, x_{i_0}]$ that is uncovered. A similar argument holds if $\Delta = d_n$. □

In the next definition we transform x into an instance x' by pushing sensors away from $\frac{1}{2}$, so that each internal gap between sensors is of equal width.

Definition 1. *For a given instance x, let k be sensor whose location is closest to $1/2$. Then we define the* stretched *instance x' of x as follows:*

$$x'_i = \begin{cases} (1 - r_k) - (\lceil n/2 \rceil - i)\Delta & i \le \lceil n/2 \rceil\,, \\ (1 - r_k) + (i - \lceil n/2 \rceil)\Delta & i > \lceil n/2 \rceil\,. \end{cases}$$

See Figure 7 in Appendix C for an illustration.

Observation 4. *Let x' be a stretched instance of x. Then $\left|\left\{i : x'_i \le \frac{1}{2}\right\}\right| = \lceil n/2 \rceil$ and $\left|\left\{i : x'_i > \frac{1}{2}\right\}\right| = \lfloor n/2 \rfloor$.*

Lemma 7. *Let x' be the stretched instance of x. Then,* $\text{Opt}_0(x') = \text{Opt}_0(x)$ *and* $\text{RR}'(x') \le \text{RR}'(x)$.

Proof. First, by construction, the internal gaps in x' are of length Δ and $d'_0, d'_1 \le \Delta$. Thus, by Lemma 6, $\text{Opt}_0(x') = \text{Opt}_0(x)$. By Lemma 5 we know that the sensors moved away from $\frac{1}{2}$, hence $\sum_i r'_i \ge \sum_i r_i$ and $\text{RR}'(x') \le \text{RR}'(x)$. □

Now we are ready to bound $\text{RR}(x)$.

Lemma 8. $\mathrm{RR}'(x) \geq \frac{2}{3}\textsc{Opt}_0(x)$, *for every instance* $I = (x, \bar{1})$ *of* RadSC, *where sensors may be located outside* $[0, 1]$.

Proof. First by Lemma 7 we may assume that the instance is stretched.

Suppose that n is even. Then since x is a stretched instance, it must be the case that exactly half of the sensors lie to the left of $1/2$, and exactly half lie to the right. Hence,

$$\begin{aligned}
\overline{r} \triangleq \frac{1}{n}\sum_{i=1}^{n} r_i &= \frac{1}{n}\left[\sum_{j=0}^{n/2-1}(r_{n/2} + j\varDelta) + \sum_{j=0}^{n/2-1}(r_{n/2+1} + j\varDelta)\right] \\
&\geq \frac{1}{n}\left[\frac{n}{2}\cdot r_{n/2} + \varDelta\binom{n/2}{2} + \frac{n}{2}\cdot r_{n/2+1} + \varDelta\binom{n/2}{2}\right] \\
&= \frac{r_{n/2} + r_{n/2+1}}{2} + \frac{2\varDelta}{n}\binom{n/2}{2} \\
&\geq \frac{1+\varDelta}{2} + \frac{\varDelta(n-2)}{4} \\
&= \frac{1}{2} + \frac{n\varDelta}{4}\ ,
\end{aligned}$$

where we have used the fact that since the sequence is stretched $r_{n/2} + r_{n/2+1} \geq 1 + \varDelta$. Furthermore, since $n\varDelta \geq 1$, it now follows that

$$\frac{\mathrm{RR}'(x)}{\textsc{Opt}_0(x)} = \frac{n/\overline{r}}{2/\varDelta} \geq \frac{n\varDelta}{1 + \frac{n\varDelta}{2}} = \frac{1}{\frac{1}{n\varDelta} + \frac{1}{2}} \geq \frac{2}{3}\ .$$

If n is odd, then without loss of generality we can assume that there are exactly $\frac{n+1}{2}$ sensors to the left of $1/2$, and exactly $\frac{n-1}{2}$ to the right. Then

$$\begin{aligned}
\overline{r} &\geq \frac{1}{n}\left[\sum_{j=0}^{(n-1)/2}(r_{(n+1)/2} + j\varDelta) + \sum_{j=0}^{(n-3)/2}(r_{(n+3)/2} + j\varDelta)\right] \\
&\geq \frac{1}{n}\left[\frac{n+1}{2}\cdot r_{(n+1)/2} + \varDelta\binom{(n+1)/2}{2} + \frac{n-1}{2}\cdot r_{(n+3)/2} + \varDelta\binom{(n-1)/2}{2}\right] \\
&= \frac{r_{(n+1)/2} + r_{(n+3)/2}}{2} + \frac{r_{(n+1)/2} - r_{(n+3)/2}}{2n} + \frac{\varDelta}{n}\cdot\frac{(n-1)^2}{4} \\
&\geq \frac{1+\varDelta}{2} + \frac{\varDelta}{n}\cdot\frac{(n-1)^2}{4} \\
&= \frac{1}{2} + \varDelta\frac{n^2+1}{4n}\ .
\end{aligned}$$

We have two cases. If $r_1 \geq 1$, then there are $n-1$ gaps of size $\varDelta$, as well as one gap of size at most $\varDelta/2$. Since the gaps cover the entire interval, we have that $(n-1)\varDelta + \frac{\varDelta}{2} \geq 1$. It follows that $n\varDelta \geq \frac{2n}{2n-1}$. Thus, we can demonstrate the same bound, since

$$\frac{\mathrm{RR}'(x)}{\textsc{Opt}_0(x)} = \frac{n/\overline{r}}{2/\varDelta} = \frac{n\varDelta}{1 + \frac{(n^2+1)\varDelta}{4n}} = \frac{1}{\frac{1}{n\varDelta} + \frac{1}{2} + \frac{1}{2n^2}} \geq \frac{2n^2}{3n^2 - n + 1} > \frac{2}{3}\ .$$

Finally, we consider the case where $r_1 < 1$. For some $\epsilon \in (0, \Delta/2]$, we can set $r_{(n+1)/2} = \frac{1}{2} + \epsilon$. Since sensors $(n+1)/2$ and $(n+3)/2$ are of distance Δ from one another, it follows that

$$r_{\frac{n+3}{2}} - r_{\frac{n+1}{2}} = \left(\frac{1}{2} + \Delta - \epsilon\right) - \left(\frac{1}{2} + \epsilon\right) = \Delta - 2\epsilon \ .$$

Moreover, we will show that $\epsilon \leq \Delta/4$, and thus $r_{(n+3)/2} - r_{(n+1)/2} \geq \Delta/2$. To see this, note first that it follows from the definition of a stretch sequence and the assumption that $r_1 < 1$ that $r_1 = r_{(n+1)/2} + \Delta(n-1)/2$ and $r_2 = r_{(n+3)/2} - \Delta(n-3)/2$. Hence their difference is

$$r_1 - r_n = \left(r_{\frac{n+1}{2}} + \frac{\Delta(n-1)}{2}\right) - \left(r_{\frac{n+3}{2}} + \frac{\Delta(n-3)}{2}\right) = r_{\frac{n+1}{2}} - r_{\frac{n+3}{2}} + \Delta = 2\epsilon \ .$$

However since $1 - \delta/2 \leq r_n \leq r_1 < 1$, it must the case that $r_1 - r_n \leq \Delta/2$, and this implies that $\epsilon \leq \Delta/4$.

Finally, a computation similar to the one above reveals that

$$\begin{aligned}\overline{r} \geq \frac{r_{\frac{n+1}{2}} + r_{\frac{n+3}{2}}}{2} + \frac{r_{\frac{n+1}{2}} - r_{\frac{n+3}{2}}}{2n} + \frac{\Delta}{n}\frac{(n-1)^2}{4} &\geq \frac{1+\Delta}{2} - \frac{\Delta}{4n} + \frac{\Delta}{n}\frac{(n-1)^2}{4} \\ &= \frac{1}{2} + \frac{n\Delta}{4} \ .\end{aligned}$$

As this is the same bound that we obtained in the even case, we similarly achieve the same 2/3 bound. □

4 Set Radius Strip Cover

In this section we present an optimal $O(n^2 \log n)$-time algorithm for the RadSC problem. Recall that in RadSC we may only set the radii of the sensors since all the activation times must be set to 0. More specifically, we assign non-zero radii to a subset of the sensors which we call *active*, while the rest of the sensors get $\rho_i = 0$ and do not participate in the cover.

Given an instance (x, b), a radial assignment ρ is called *proper* if the following conditions hold:

1. Every sensor is either inactive, or exhausts its battery by time T, where T is the lifetime of ρ. That is, $\rho_i \in \{0, b_i/T\}$,
2. No sensor's coverage is superfluous. That is, for every active sensor i there is a point $u_i \in [0,1]$ such that $u_i \in [x_i - \rho_i, x_i + \rho_i]$ and $u_i \notin [x_k - \rho_k, x_k + \rho_k]$, for every active $k \neq i$.

Lemma 9. *There exists a proper optimal assignment for every* RadSC *instance* $I = (x, b)$.

Proof. Let ρ be an optimal assignment for I with lifetime T. We first define the assignment $\rho' = b/T$ and show that it is feasible. Since ρ has lifetime T, any point $u \in [0,1]$ is covered by some sensor i throughout the time interval $[0,T]$. It follows that $\rho_i \le b_i/T = \rho'_i$. Hence, $u \in [x_i - \rho'_i, x_i + \rho'_i]$, and thus ρ' has lifetime T. Next, we construct an assignment ρ''. Initially, $\rho'' = \rho'$. Then starting with $i = 1$, we set $\rho''_i = 0$ as long as ρ'' remains feasible. Clearly, $\rho''_i \in \{0, b_i/T\}$. Furthermore, for every sensor i there must be a point $u_i \in [x_i - \rho''_i, x_i + \rho''_i]$ such that $u_i \notin [x_k - \rho''_k, x_k + \rho''_k]$, for every active $k \ne i$, since otherwise i would have been deactivated. Hence, ρ'' is a proper assignment with lifetime T, and is thus optimal. □

Given a proper optimal solution, we add two dummy sensors, denoted 0 and $n+1$, with zero radii and zero batteries at 0 and at 1, respectively. The dummy sensors are considered active. We show that the optimal lifetime of a given instance is determined by at most two active sensors.

Lemma 10. *Let T be the optimal lifetime of a given* RadSC *instance $I = (x,b)$. There exist two sensors $i, k \in \{0, \ldots, n+1\}$, where $i < k$, such that $T = \frac{b_k + b_i}{x_k - x_i}$.*

Proof. Let ρ be the proper optimal assignment, whose existence is guaranteed by Lemma 9. We claim that there exist two neighboring active sensors i and k, where $i < k$, such that $\rho_i + \rho_k = x_k - x_i$. The lemma follows, since $\rho_i = b_i/T$ and $\rho_k = b_k/T$.

Observe that if $\rho_i + \rho_k < x_k - x_i$, for two neighboring active sensors i and k, then there is a point in the interval (x_i, x_k) that is covered by neither i and k, but is covered by another sensor. This means that either i or k is redundant, in contradiction to ρ being proper. Hence, $\rho_i + \rho_k \ge x_k - x_i$, for every two neighboring active sensors i and k.

Let $\alpha = \min\left\{\frac{\rho_k + \rho_i}{x_k - x_i} : i, k \text{ are active}\right\}$. If $\alpha = 1$, then we are done. Otherwise, we define the assignment $\rho' = \rho/\alpha$. ρ' is feasible since $\rho'_i + \rho'_k = \frac{1}{\alpha}(\rho_i + \rho_k) \ge x_k - x_i$, for every two neighboring active sensors i and k. Furthermore, the lifetime of ρ' is αT, in contradiction to the optimality of ρ. □

Lemma 10 implies that there are $O(n^2)$ possible lifetimes. This leads to an algorithm for solving RadSC.

Theorem 2. *There exists an $O(n^2 \log n)$-time algorithm for solving* RadSC.

5 Strip Cover

Theorem 1 can be extended to the Strip Cover problem.

Theorem 3. RoundRobin *is a $\frac{3}{2}$-approximation algorithm for* Strip Cover.

We also show that there is a gap between the lifetime of the optimal solutions of OnceSC and Strip Cover.

Lemma 11. *There exists an instance (x,b) for which the ratio between the optimal value for* Strip Cover *and the optimal value of* OnceSC *is $\frac{7}{6}$.*

6 Discussion and Open Problems

We have shown that RoundRobin, which is perhaps the simplest possible algorithm, has a tight approximation ratio of $\frac{3}{2}$ for both OnceSC and Strip Cover. We also showed that OnceSC is NP-hard, but it remains to be seen whether the same is true for Strip Cover. Future work may include finding algorithms with better approximation ratios for either problem. However, we have eliminated duty cycle algorithms as candidates. Observe that both OnceSC and TimeSC are NP-hard, while RadSC can be solved in polynomial time. This suggests that hardness comes from setting the activation times.

This problem setting is rich, in that there are many variations in the setup that can alter the resulting analysis dramatically. In this paper we have assumed that the battery charges dissipate in direct inverse proportion to the assigned sensing radius (e.g. $\tau = b/\rho$). It is natural to suppose that an exponent could factor into this relationship, so that, say, the radius drains in quadratic inverse proportion to the sensing radius (e.g. $\tau = b/\rho^2$).

Also of interest would be to expand the scope of the problem to higher dimensions. Even before moving both the sensor locations and the region being covered to the plane, one might consider moving one but not the other. This yields two different problems: 1) covering the line with sensors located in the plane; and 2) covering a region of the plane with sensors located on a line.

References

1. G. Aloupis, J. Cardinal, S. Collette, S. Langerman, D. Orden, and P. Ramos. Decomposition of multiple coverings into more parts. *Discrete & Computational Geometry*, 44(3):706–723, 2010.
2. A. Bar-Noy and B. Baumer. Maximizing network lifetime on the line with adjustable sensing ranges. In *ALGOSENSORS*, pages 28–41, 2011.
3. A. Bar-Noy, T. Brown, M.P. Johnson, and O. Liu. Cheap or Flexible Sensor Coverage. In *DCOSS*, pages 245–258. Springer, 2009.
4. A.L. Buchsbaum, A. Efrat, S. Jain, S. Venkatasubramanian, and K. Yi. Restricted strip covering and the sensor cover problem. In *SODA*, pages 1056–1063, 2007.
5. E. Chambers, S. Fekete, H.F. Hoffmann, D. Marinakis, J. Mitchell, V. Srinivasan, U. Stege, and S. Whitesides. Connecting a set of circles with minimum sum of radii. In *WADS*, pages 183–194. Springer, 2011.
6. M. Gibson and K. Varadarajan. Decomposing coverings and the planar sensor cover problem. In *FOCS*, pages 159–168. IEEE Computer Society, 2009.
7. N. Lev-Tov and D. Peleg. Polynomial time approximation schemes for base station coverage with minimum total radii. *Computer Networks*, 47(4):489–501, 2005.
8. J. Pach. Covering the plane with convex polygons. *Discrete & Computational Geometry*, 1(1):73–81, 1986.
9. J. Pach and G. Tóth. Decomposition of multiple coverings into many parts. *Computational Geometry*, 42(2):127–133, 2009.

A Set Once Hardness Result

In this section we show that OnceSC is NP-hard. This is done using a reduction from Partition.

Theorem 4. OnceSC *is NP-hard.*

Proof. Let $Y = \{y_1, \ldots, y_n\}$ be a given instance of Partition, and define $B = \frac{1}{2}\sum_{i=1}^{n} y_i$. We create an instance of OnceSC by placing n sensors with battery y_i at $\frac{1}{2}$, and two additional sensors equipped with battery B at $\frac{1}{6}$ and $\frac{5}{6}$, respectively. That is, the instance of OnceSC consists of sensor locations $x = (\frac{1}{6}, \underbrace{\frac{1}{2}, \ldots, \frac{1}{2}}_{n}, \frac{5}{6})$ and batteries $b = (B, y_1, \ldots, y_n, B)$. We show that $Y \in$ Partition if and only if the maximum possible lifetime of $8B$ is achievable for the OnceSC instance.

First, suppose $Y \in$ Partition, hence there exist two non-empty disjoint subsets $Y_0, Y_1 \subseteq Y$, such that $Y_0 \cup Y_1 = Y$, and $\sum_{y \in Y_0} y = B = \sum_{y \in Y_1} y$. Schedule the sensors in Y_0 to iteratively cover the region $[\frac{1}{3}, \frac{2}{3}]$. Since all of these sensors are located at $\frac{1}{2}$, this requires that each sensor's radius be set to $\frac{1}{6}$, i.e. $\rho_{i+1} = \frac{1}{6}$, for every $i \in Y_0$. Since the sum of their batteries is B, this region can be covered for exactly $6B$ time units. With the help of the additional sensors located at $\frac{1}{6}$ and $\frac{5}{6}$, whose radii are also set to $\rho_1 = \rho_{n+2} = \frac{1}{6}$, the sensors in Y_0 can thus cover $[0, 1]$ for $6B$ time units (see Figure 3 for an example). Next, the sensors in Y_1 can cover $[0, 1]$ for an additional $2B$ time units, since they all require a radius of $\rho_{i+1} = \frac{1}{2}$, for every $i \in Y_1$. Thus, the total lifetime is $8B$.

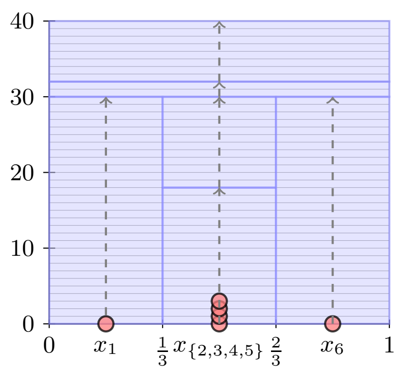


Fig. 3. Proof of NP-hardness. $Y = \{1, 2, 3, 4\}$ is a given instance of Partition, and $(x, b) = \big((\frac{1}{6}, \frac{1}{2}, \ldots, \frac{1}{2}, \frac{5}{6}), (5, 1, 2, 3, 4, 5)\big)$ is the translated OnceSC instance.

Now suppose that for such a OnceSC instance, the lifetime of $8B$ is achievable. Since the maximum possible lifetime is achievable, no coverage can be wasted in the optimal schedule. In this case the radii of the sensors at $\frac{1}{6}$ and $\frac{5}{6}$

must be exactly $\frac{1}{6}$, since otherwise, they would either not reach the endpoints $\{0,1\}$, or extend beyond them. Moreover, due the fact that all of the other sensors are located at $\frac{1}{2}$, and their coverage is thus symmetric with respect to $\frac{1}{2}$, it cannot be the case that sensor 1 and sensor $n+2$ are active at different times. Thus, the solution requires a partition of the sensors located at $\frac{1}{2}$ into two groups: the first of which must work alongside sensors 1 and $n+2$ with a radius of $\frac{1}{6}$ and a combined lifetime of $6B$; and the second of which must implement RoundRobin for a lifetime of $2B$. The batteries of these two partitions form a solution to Partition. □

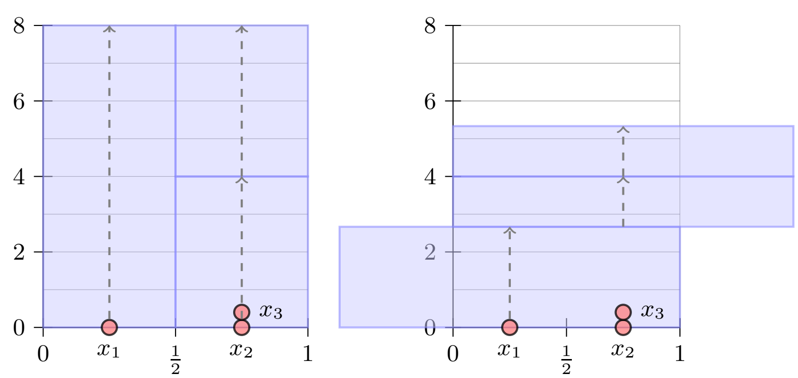
B Duty Cycle Algorithms

In this paper we analyzed the RoundRobin algorithm in which each sensor works alone. One may consider a more general version of this approach, where a schedule induces a partition of the sensors into sets, or *shifts*, and each shift works by itself. In RoundRobin each shift consists of one active sensor. We refer to such an algorithm as a *duty cycle* algorithm.

In this section we show that, in the worst case, no duty cycle algorithm performs better than RoundRobin. More specifically, we show that the approximation ratio of any duty cycle algorithm is at least $\frac{3}{2}$.

Lemma 12. *The approximation ratio of any duty cycle algorithm is at least* $\frac{3}{2}$.

Proof. Consider an instance where $x = (\frac{1}{4}, \frac{3}{4}, \frac{3}{4})$ and $b = (2, 1, 1)$. An optimal solution is obtained by assigning $\rho_1 = \rho_2 = \rho_3 = \frac{1}{4}$, $\tau_1 = \tau_2 = 0$ and $\tau_3 = 4$. That is, sensor 1 covers the interval $[0, 0.5]$ for 8 time units, sensors 2 covers $[0.5, 1]$ until time 4, and sensors 3 covers $[0.5, 1]$ from time 4 to 8. This solution is optimal in that it achieves the maximum possible lifetime of $8 = 2\sum_i b_i$.



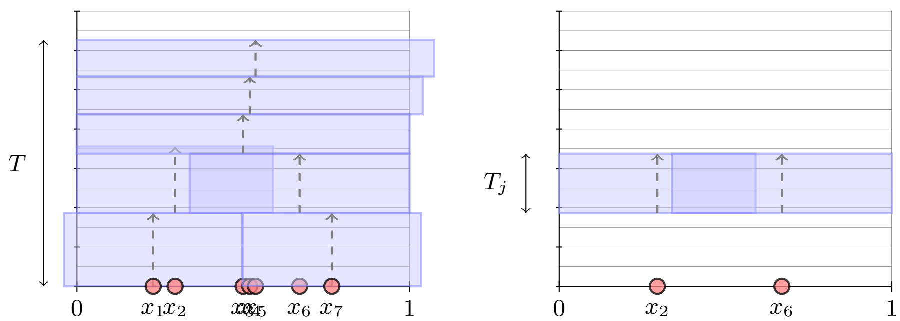
(a) Opt$(x, b) = 8$ (b) RoundRobin$(x, b) = 16/3$

Fig. 4. Best schedule vs. best duty cycle schedule. Here $x = (\frac{1}{4}, \frac{3}{4}, \frac{3}{4})$ and $b = (2, 1, 1)$.

On the other hand, the best duty cycle algorithm is RoundRobin, which achieves a lifetime of $16/3$ time units. (The shifts $\{1, 2\}$ and $\{3\}$ would also result in a lifetime of $16/3$ time units.) Both schedules are shown in Figure 4. □

C Additional Figures

Figure 5 illustrates the procedure of cutting an optimal schedule into strips.



(a) An optimal schedule S (b) One strip of that schedule S_j

Fig. 5. Cutting an optimal schedule into strips. Note that coverage overlaps may occur in both the horizontal and vertical directions in the optimal schedule, but only horizontally in a strip.

Figure 6 illustrates the reduction from a non-uniform battery instance in a particular strip, to a unit battery instance.

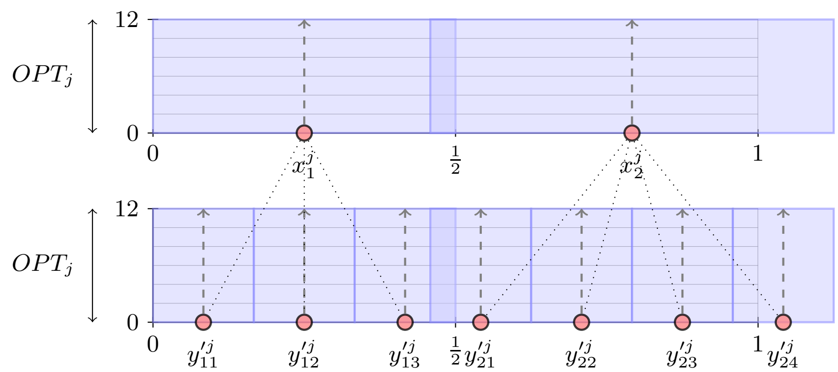


Fig. 6. Reduction of non-uniform battery instance I_j to uniform battery instance I'_j: At the top, $I_j = \big((\frac{1}{4}, \frac{19}{24}), (3, 4)\big)$, while at the bottom, $I'_j = \big((\frac{1}{12}, \frac{3}{12}, \frac{5}{12}, \frac{13}{24}, \frac{17}{24}, \frac{21}{24}, \frac{25}{24}), \bar{1}\big)$.

Figure 7 is an illustration of how unit battery instances are stretched.

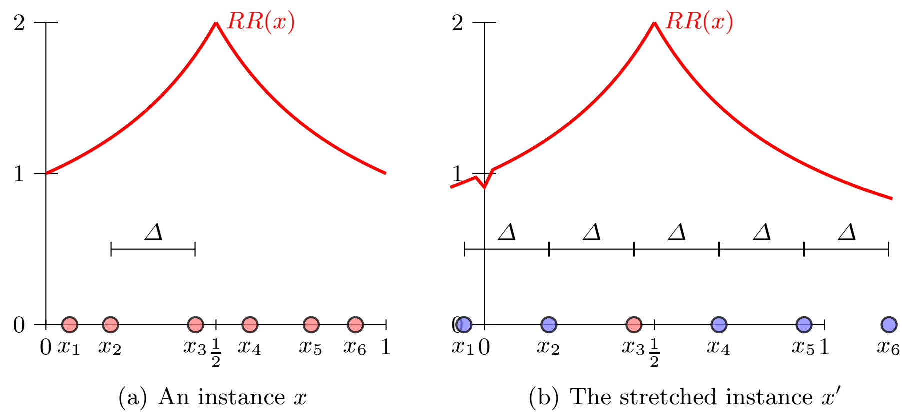


Fig. 7. Transformation of instance x to stretched instance x'. The sensor closest to $\frac{1}{2}$ (x_3) remains in place, while the other sensors are placed at increasing intervals of Δ away from x_3.

D Omitted Proofs

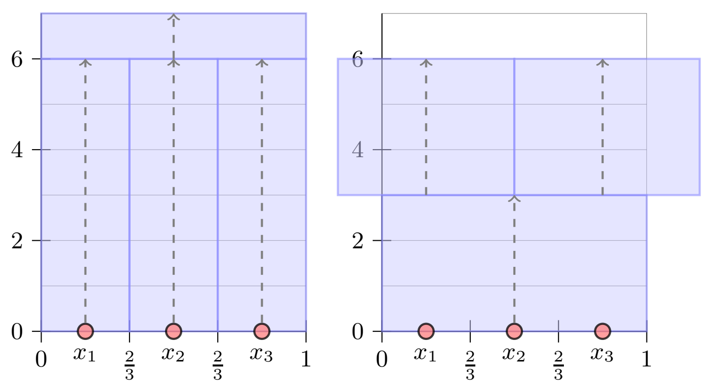
Proof (of Theorem 2). First if $n = 1$, then $\rho_1 \leftarrow r_1 \triangleq \max(x_1, 1-x_1)$ and we are done. Otherwise, let $T_{ik} \leftarrow \frac{b_k+b_i}{x_k-x_i}$, for every $i, k \in \{0, \dots, n+1\}$ such that $i < k$. After sorting the set $\{T_{ik} : i < k\}$, perform a binary search to find the largest potentially feasible lifetime. The feasibility of candidate T_{ik} can be checked using the assignment $\rho^{ik}_\ell \leftarrow b_\ell / T_{ik}$, for every sensor ℓ.

There are $O(n^2)$ candidates, each takes $O(1)$ to compute, and sorting takes $O(n^2 \log n)$ time. Checking the feasibility of a candidate takes $O(n)$ time, and thus the binary search takes $O(n \log n)$. Hence, the overall running time is $O(n^2 \log n)$. □

Proof (of Lemma 11). Consider the instance with three sensors where $x = (\frac{1}{6}, \frac{1}{2}, \frac{5}{6})$ and $b = (1, 1.5, 1)$.

In Strip Cover all can work together with radii $\frac{1}{6}$ for 6 time units, and then the second sensor can survive for 1 more time units with by adjusting its radius to $\frac{1}{2}$. Hence we get a lifetime of 7 which is the best possible.

On the other hand, in OnceSC sensor 2 has two radial options, $\frac{1}{6}$ and $\frac{1}{2}$. If the radius of sensor 2 is set to $\frac{1}{6}$, then all sensors work together for 6 time units. Otherwise, if its radius is set to $\frac{1}{2}$ then it works alone for 3 time units. The other sensors can work for an additional 3 units, and once again we get 6 units of lifetime. See Figure 8. □



(a) Strip Cover$(x, b) = 7$ (b) OnceSC$(x, b) = 16/3$

Fig. 8. An optimal OnceSC solution is at least a $\frac{7}{6}$ approximation of an optimal Strip Cover solution.